 <p><b>MOTION IMAGERY STANDARDS BOARD</b></p> <p><b>STANDARD</b></p> <p><b>Motion Imagery Metadata (MIMD): Model</b></p>	<p><b>MISB ST 1903.1</b></p> <p><b>25 June 2020</b></p>
---	---

## 1 Scope

The Motion Imagery Metadata (MIMD) documents are composed of the MIMD Modeling Rules, MIMD Model, and Model-to-KLV Transmutation Instructions. MISB ST 1901 [1] defines the modelling rules which are a subset of standardized UML class diagrams specifically minimized to enable transforming, or transmuting, class instances to KLV. MISB ST 1902 [2] defines the Model-to-KLV Transmutation Instructions for constructing bandwidth efficient KLV structures from the model.

This standard and its supporting documents defines the MIMD model, which is a UML model organized as a hierarchy of information including metadata for temporal, platform, payload, sensor, command, automated processes, exploitation, security, and more. This document, in concert with MISB ST 1901 and MISB ST 1902, provides a suite of standards embodying the MIMD architecture.

## 2 References

- [1] MISB ST 1901.1 Motion Imagery Metadata (MIMD): Modeling Rules, Jun 2020.
- [2] MISB ST 1902.1 Motion Imagery Metadata (MIMD): Model-to-KLV Transmutation Instructions, Jun 2020.
- [3] MISB ST 0107.4 KLV Metadata in Motion Imagery, Feb 2019.
- [4] MISB ST 1904.1 Motion Imagery Metadata (MIMD): Base Attributes, Jun 2020.
- [5] MISB ST 1603.2 Time Transfer Pack, Oct 2017.
- [6] MISB ST 1906.1 Motion Imagery Metadata (MIMD): Staging System, Jun 2020.
- [7] MISB ST 1905.1 Motion Imagery Metadata (MIMD): Platform, Jun 2020.
- [8] MISB MISP-2020.1: Motion Imagery Handbook, Oct 2019.
- [9] MISB ST 0603.5 MISP Time System and Timestamps, Oct 2017.
- [10] MISB ST 0601.16 UAS Datalink Local Set, Oct 2019.
- [11] DOD Manual 5200.01 Volume 2.Incorporating Change 3, Effective May 14, 2019 DoD Information Security Program: Marking of Classified Information, 19 Mar 2013.
- [12] W3C Recommendation - OWL Web Ontology Language, 2009.

- [13] Assistant Secretary of Defense for Command, Control, Communications and Intelligence  
Global Positioning Standard Positioning Service Performance Standard 4th Ed, Sept 2008.
- [14] IEEE Standard 1588-2002.1 IEEE Standard for a Precision Clock Synchronization Protocol  
for Networked Measurement and Control Systems, 2002.
- [15] IEEE Standard 1588-2008.2 IEEE Standard for a Precision Clock Synchronization Protocol  
for Networked Measurement and Control Systems, 2008.
- [16] IETF RFC 1305 Ver. 3 .3 Network Time Protocol, 1992.
- [17] IETF RFC 5905 Ver. 4 Network Time Protocol, Jun 2010.
- [18] IRIG Standard 200-04 Telecommunications and Timing Group/Range Commanders  
Council, Sep 2004.

### 3 Revision History

Revision	Date	Summary of Changes
ST 1903.1	6/25/2020	<ul style="list-style-type: none"> <li>• Added groupIdRange to support mimdId groups</li> <li>• Added algorithm list to identify algorithms in use throughout the MIMD model</li> <li>• Added ontologies list to identify ontology information as needed throughout the MIMD model</li> <li>• Added track list to support object tracking</li> <li>• Restructured to enable auto-generation of Model Class and Model Enumeration sections</li> <li>• Added requirements 14, 15, 16, and 17</li> </ul>

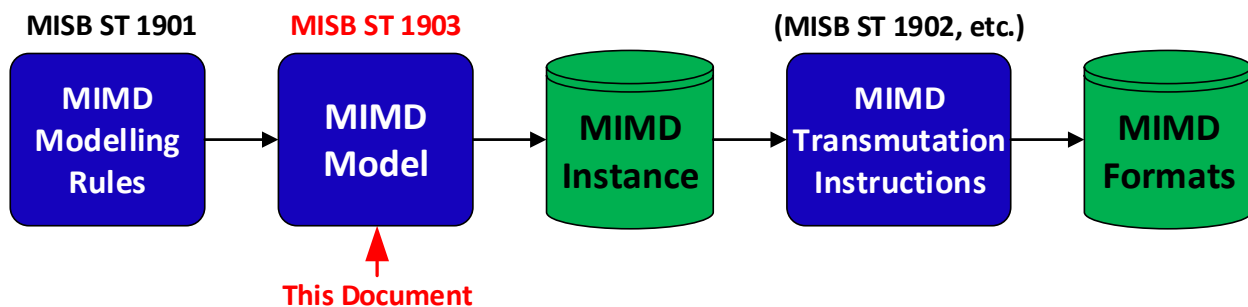
### 4 Acronyms, Terms, Definitions

<b>DOD</b>	Department of Defense
<b>GPS</b>	Global Positioning System
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IETF</b>	Internet Engineering Task Force
<b>IRIG-A</b>	Inter-range Instrumentation Group version A
<b>IRIG-B</b>	Inter-range Instrumentation Group version B
<b>KLV</b>	Key Length Value
<b>LS</b>	Local Set
<b>MIMD</b>	Motion Imagery Metadata
<b>MISB</b>	Motion Imagery Standards Board
<b>MISP</b>	Motion Imagery Standards Profile
<b>NTP</b>	Network Time Protocol
<b>PTP</b>	Precision Time Protocol
<b>ST</b>	Standard
<b>ROC_Unknown</b>	Report-on-Change Unknown state (see MISB ST 1901)
<b>UML</b>	Unified Modelling Language
<b>XML</b>	eXtensible Markup Language

## 5 Introduction

Motion Imagery metadata facilitates managing, discovering, and providing contextual information for exploiting Motion Imagery. To date the MISB has published numerous metadata standards to address the needs of the ISR Motion Imagery community. New applications demand additions to these standards as well as the creation of new ones, which warrants the MISB to adopt a more holistic approach to metadata in general. The purpose of the MIMD suite of standards is to provide a framework that consolidates and organizes information into a singular, unified metadata model, thereby serving many use cases. This MIMD model then transmutes into different formats, such as KLV or XML.

The MIMD standards address different aspects of the MIMD system, which defines the processes for generating efficient KLV (and other future formats) from a MIMD model. Figure 1 identifies the five components of the MIMD system: MIMD Modeling Rules, MIMD Model, MIMD Instance, MIMD Transmutation Instructions, and MIMD Formats. The blue items represent MISB standards and the green items represent implementation data.



**Figure 1: MIMD System Components**

The MIMD Modeling Rules comprise a subset of the UML rules for building the MIMD Model. These rules enable transforming, or transmuting, MIMD Instances into MIMD Formats by the MIMD Transmutation Instruction documents.

The MIMD Model is a UML data model following the MIMD Modeling Rules. The Model contains classes, class attributes and defines relationships between classes. MISB ST 1903 and its supporting documents define the MIMD Model.

A MIMD Instance is a data structure implementation of the MIMD Model. While a class defines the allowed contents and rules, an instance is a specific implementation of the class. In other words, a class is the blueprint; the instance is the actual data.

The MIMD KLV Transmutation Instructions provides the “algorithm” for converting a MIMD Model Instance into MIMD Formats. Currently, the only MIMD Format is KLV. MISB ST 1902 defines the MIMD Model-to-KLV Transmutation Instructions that maps instances and their relationships to hierarchical KLV structures (e.g., Local Sets and Packs). Additionally, these instructions define methods for the “special” model functions for KLV-unique encodings (e.g., temporal compression). The MISB intends to develop other transmutation instructions for other formats such as XML or JSON as needed.

The MIMD Formats are the final product of the MIMD system. The MIMD KLV Format is a bit-efficient implementation of the MIMD Instance data. Systems generate custom subsets of the MIMD KLV by selecting appropriate classes and class attributes for their system and following the MIMD Model-to-KLV Transmutation Instructions and requirements. Systems that create metadata and transmute it to MIMD Formats are producers. Systems that decode MIMD Formats and process the metadata are receivers.

## 6 MIMD Model Overview

The MIMD Model is a set of hierarchical UML class definitions based on the rules defined in MISB ST 1901. MISB ST 1901 also defines MIMD Packets and the use of the Report-on-Change (MISB ST 0107 [3]) when transmitting MIMD Model data.

The MIMD Model defines the allowed structure and relationships of classes that systems may use to report their metadata. At run-time, systems create instances of the classes they need from the MIMD Model definition; systems therefore will, in general, use a subset of the full MIMD Model.

### 6.1 Model Architecture

The Model Architecture is hierarchical with the MIMD class defining the root class. All classes in the model are children (or children of children) of the MIMD class and either provide an object's (or concept's) modelling information via classes and attributes, or support model infrastructure, or both. Descriptions in MISB ST 1903 and supporting documents provide details of each class's modeling information.

The Model Architecture includes “infrastructures” which combine classes and rules to support whole-model objectives, such as proper timing and security. The following sections provide details on MIMD Model infrastructures: Association Relationships, Timing, Staging, Standard Deviations with Cross Correlations, and Security.

#### 6.1.1 Association Relationships

Association relationships enable class instances to reference, or link, to other class instances in the MIMD Model at run-time. The model defines the allowed references which enables disparate parts of the model hierarchy to reference each other. Each MIMD Model class includes the Base Attributes (see MISB ST 1904 [4]) which include the MIMD instance Identifier (mimId). The mimId is the attribute which enables class instance associations; see MISB ST 1901 for details about association relationships. Both the timing and staging infrastructures utilize associations.

#### 6.1.2 Timing

The timing infrastructure enables producers to assign detailed timing information throughout the model. The timing infrastructure is a combination of the Timer class list and two Base Attributes: Timer Identifier (timerId) and Timer Offset (timerOffset). The Timer class list is an attribute of the MIMD class providing the timing information for any class instance in the MIMD Model. Section 7.1 lists the attributes of the Timer class. A physical Timer is a system which provides a

time source for a device within the Motion Imagery System. The Timer is typically a Time Transfer device as defined in MISB ST 1603 [5]. At a minimum there is one Timer in the list.

The Base Attributes provide a method to associate any class instance to any of the Timer instances in the Timer class list by using the timerId. Furthermore, each class instance may include an offset to the specified timer (timerOffset) to fine tune the timing of a class instance with either a positive or negative offset value. MISB ST 1901 provides details on the timerId and timerOffset attributes.

The MIMD Model timing references are hierarchical so a time reference at a parent class flows down to all children under the parent. Children use their parent's time reference by default or may specify their own time reference which flows down to their children, etc. By default, the first timer in the Timer list is the default time for the whole MIMD Model.

Figure 2 illustrates an example of the timing infrastructure in a MIMD model with two sensors each using their own GPS-INS systems. Each GPS-INS has a different quality of time measurement and either may provide time for other parts of the Motion Imagery System. The producer defines each time source with its own mimdId in the Timer class list. Each sensor's class instance specifies which time source they use, via the Timer Identification (timerId) and Timer Offset (timerOffset) attributes. All classes below the time source (in the class hierarchy) use the same time source as their parents, so only one specification sets the timing for a group of classes (this saves on bandwidth). Child classes may override or add offsets to tune their timing needs.

Figure 2 shows the MIMD class with a Timer class list containing Timer instances T1, T2 and T3, along with a single Platform instance. The Platform instance has a Sensor class list with each sensor having a class for the Scene and View Port. The Timers are color coded to show which timer each class uses. To aid the illustration the black boxes represent the list. The MIMD and Platform use the default timer T1. The circle with T2 on the Sensor 1 class indicates Sensor 1 has set the Timer Identifier to T2's id. This means the class Sensor 1 and its children all use T2 for their timing information. Similarly, Sensor 2 and its children use T3 for their timing.

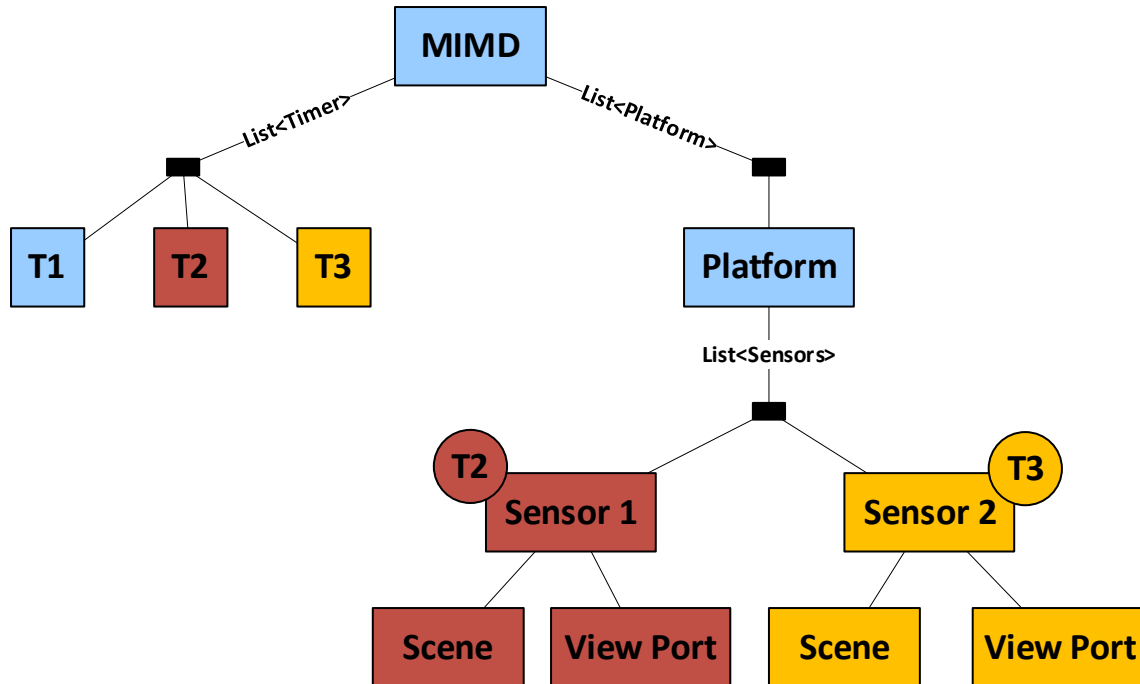


Figure 2: Timer Example

### 6.1.3 Staging System

The Staging System infrastructure enables producers to provide position, orientation, and kinematics information to any class within the model. Furthermore, producers use the Staging System to link or relate the position and orientation information from different classes together, if desired. The Staging System is a collection of classes and utilizes Association Relationships to link different stages together. MISB ST 1906 [6] provides background and modeling information on the MIMD Staging System.

### 6.1.4 Standard Deviations and Cross Correlations

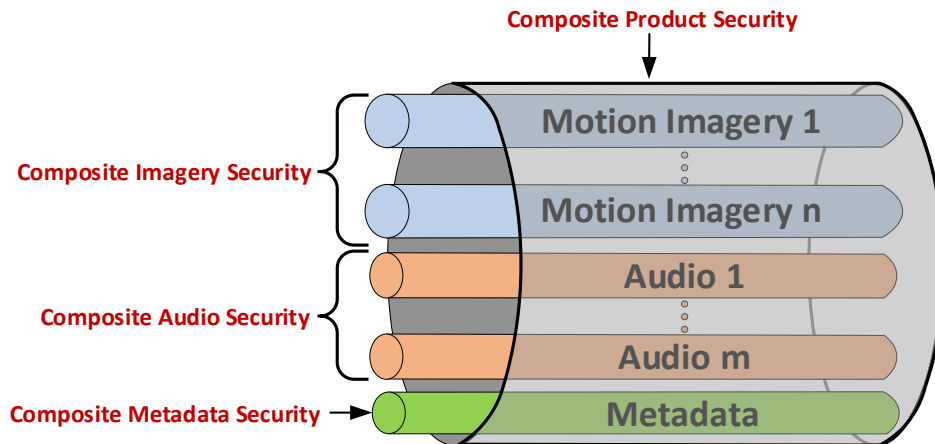
The Standard Deviation and Cross Correlations infrastructure enables producers to include error estimates along with the metadata. MISB ST 1904 provides details on Standard Deviation and Cross Correlations.

### 6.1.5 Security

The Security infrastructure enables producers to state and manage security markings for all essences contributing to a Motion Imagery Product, where a Motion Imagery Product consists of the essence types Motion Imagery, audio, and metadata. The security infrastructure includes composite security markings and individual security markings. All security infrastructure relies on the Security class list, which is an attribute of the MIMD class. Each item of the Security class list defines a security marking method and the security marking. Section 7.3 lists the attributes of the Security class.

### 6.1.5.1 Composite Security Markings

Composite security markings define an overall security marking for a group of essences comprising a Motion Imagery Product. Figure 3 illustrates four composite security markings (in red).



**Figure 3: Composite Security Markings**

Composite security markings are set either as the highest-level marking of all essences in a group (as determined by the producer), or to a level higher than any single essence in the group for cases where combining essences induces a higher security level. For example, suppose a Motion Imagery Product consists of one Motion Imagery essence and a metadata essence, with an unclassified security level for the Motion Imagery essence and an unclassified security level for the metadata essence. Further, suppose the combination of the two is at the secret security level. In this case, the Composite Product Security is set at the secret security level. The method or technique of comparing security markings is not in the scope of this standard and is the purview of security standards. Any discussion in this document where the term “highest-level” security or marking means an outside entity has declared one security marking “higher” or “lower” than the other. Composite security markings are attributes within the MIMD class.

The **Composite Product Security** marking is the overall security marking for the entire Motion Imagery Product. The Composite Product Security is the only required security marking, all other composite markings default to the Composite Product Security marking, if they are not set.

The **Composite Motion Imagery Security** marking is the security marking for all the Motion Imagery essences of the Motion Imagery Product. If separating the imagery from the Motion Imagery Product this is the security marking to use for the resulting product.

The **Composite Audio Security** marking is the security marking for all the audio essences of the Motion Imagery Product. If separating the audio from the Motion Imagery Product this is the security marking to use for the resulting product.

The **Composite Metadata Security** is the overall security marking for the entire MIMD metadata instance hierarchy. If separating the metadata from the Motion Imagery Product this is the security marking to use for the resulting product.

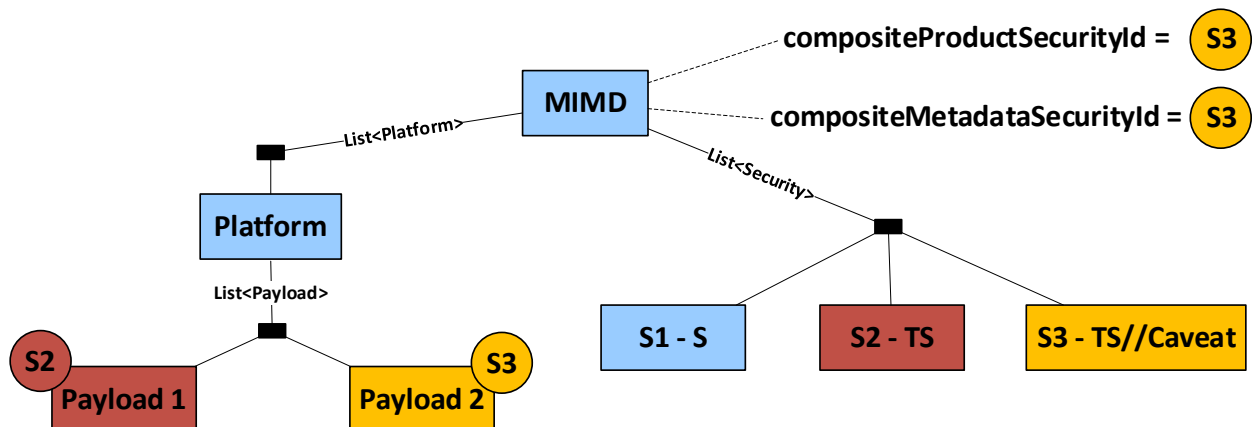
### 6.1.5.2 Individual Essence Security Markings

There are three types of individual essence security markings: Motion Imagery, audio, and metadata. Currently the MIMD Model only supports a single Motion Imagery essence, therefore the Composite Motion Imagery Security provides the individual security marking for the Motion Imagery essence. A single Motion Imagery essence may include a merging of imagery from multiple image sensors (e.g., infrared overlaid on visible light imagery). Currently the MIMD Model only supports a single audio essence, therefore, the Composite Audio Security provides the individual security marking for the audio essence.

The metadata essence contains the MIMD Model hierarchy which includes using individual, or class-by-class, security markings. Any class instance in the MIMD Model may reference any instance of the Security class list thereby providing class-based security markings. The Base Attributes include the security attribute which may reference any of the Security class instances in the Security class list. MISB ST 1904 provides details on the security attribute.

The MIMD Model security references are hierarchical, so a security reference in a parent instance becomes the default security reference for all children (and their children) under the parent. Children instances may override the default security reference with different security references, either higher-level or lower-level markings, as needed. By default, the first security class in the Security class list is the default security for the whole MIMD Model.

Figure 4 illustrates an example of the Security infrastructure in a MIMD model with two sensors. The overall, or default, security marking is the first element, S1 at the secret (S) level, of the Security list. S1 (blue) applies to the MIMD class and Platform classes. Payload 1 references the S2 (red) security marking, so the attributes of Payload 1 and its children are at the TS level. Payload 2 references the S3 (yellow) security marking, so its attributes and its children are at the TS//Caveat level.



**Figure 4: Security in MIMD Model**

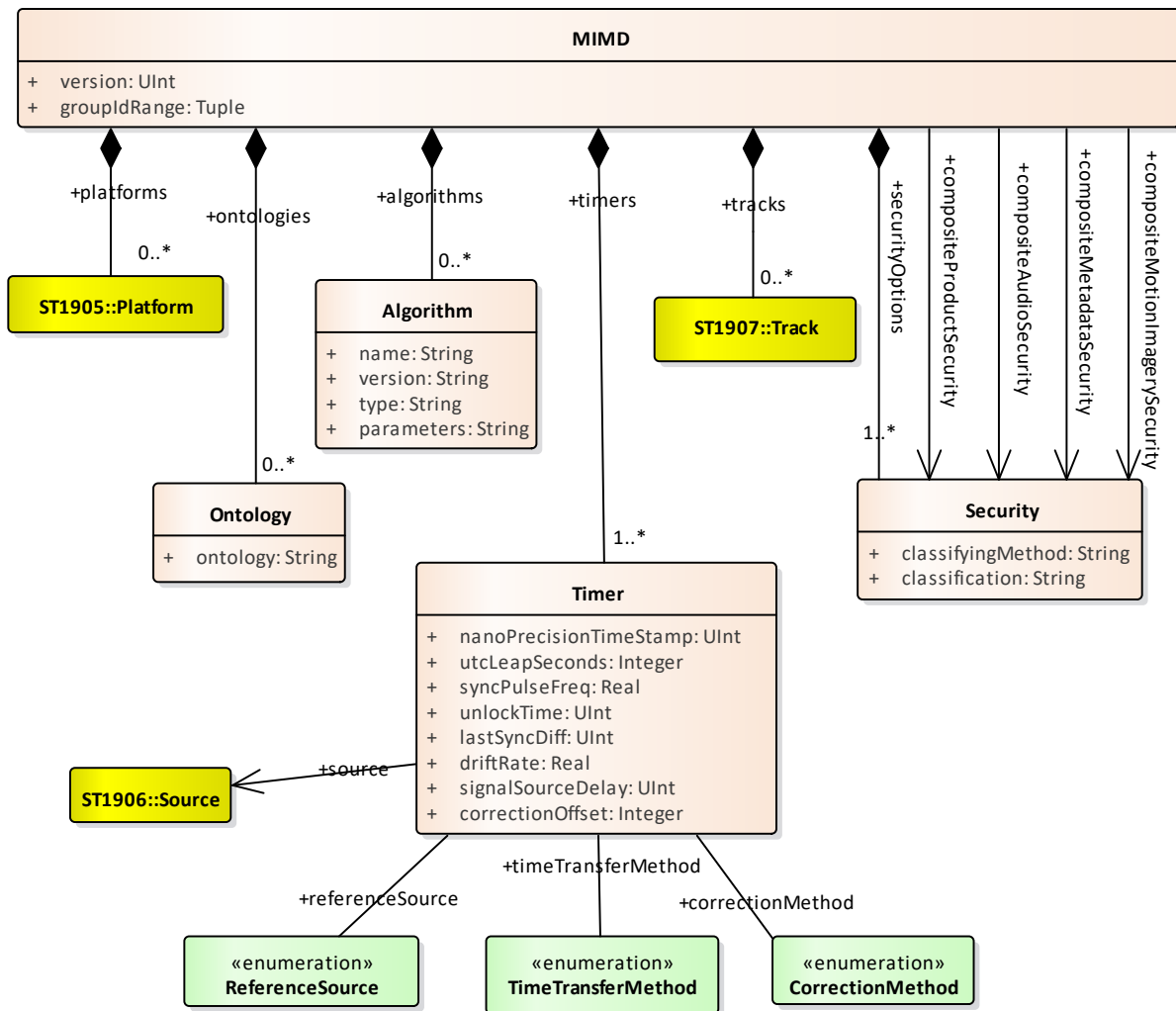
When a producer uses more than one item in the Security class list, the producer needs to specify which classification is the composite metadata security marking, as described in Section 6.1.5.1. The MIMD class diagram includes references for compositeMetadataSecurity indicating the overall security marking for all the metadata. Per Section 7.1, the MIMD class includes the compositeProductSecurity reference indicating the overall product classification marking. Figure

4 illustrates the compositeMetadataSecurity and compositeProductSecurity references both set to S3.

## 7 Model Classes

### 7.1 MIMD Class

Figure 5 shows the MIMD Model root class and its immediate children. The MIMD class provides model administrative attributes along with a Timer class list, Platform class list, and a Security class list.



**Figure 5: MIMD Model**

The Timer class list is the basis for the MIMD Model's timing infrastructure; see Section 6.1.2 for the overview of the Timing infrastructure. The Platform class list contains attributes and

classes supporting platform information; see MISB ST 1905 [7] for details the Platform class and its child classes. The Security class list provides the security details for the MIMD Model; see Section 6.1.5 for the overview of the Security infrastructure.

Per discussion in MISB ST 1901, MIMD classes support the Report-on-Change (reference the Motion Imagery Handbook [8]) and include MIMD Base Attributes (see MISB ST 1904), which use attribute identifiers 1-32.

The MIMD class contains attributes (see Table 1) for the model's version number, composite Security, a Timer class list, a Platform class list, and a Security class list.

**Table 1: MIMD Class Attributes<sup>1</sup>**

Id	Name	Type	Min	Max	Res	MLen	Units	Ref
33	version	UInt	--	--	N/A	N/A	None	7.1.1
34	compositeProductSecurity	REF<Security>	N/A	N/A	N/A	N/A	None	7.1.2
35	compositeMotionImagerySecurity	REF<Security>	N/A	N/A	N/A	N/A	None	7.1.3
36	compositeAudioSecurity	REF<Security>	N/A	N/A	N/A	N/A	None	7.1.4
37	compositeMetadataSecurity	REF<Security>	N/A	N/A	N/A	N/A	None	7.1.5
38	timers	LIST<Timer>	1	*	N/A	N/A	None	7.1.6
39	platforms	LIST<Platform>	--	--	N/A	N/A	None	7.1.7
40	securityOptions	LIST<Security>	1	*	N/A	N/A	None	7.1.8
41	groupIdRange	Tuple	N/A	N/A	N/A	2	None	7.1.9
42	algorithms	LIST<Algorithm>	--	--	N/A	N/A	None	7.1.10
43	ontologies	LIST<Ontology>	--	--	N/A	N/A	None	7.1.11
44	tracks	LIST<Track>	--	--	N/A	N/A	None	7.1.12

### 7.1.1 Attribute 33 – version

The version attribute contains the version of the MIMD Model.

The MIMD Model version for the set of MIMD documents dated June 25<sup>th</sup> 2020, is equal to “1”.

The version attribute is incremented every time there is a change to the model by the MISB. Producers need to include the version attribute in all MIMD class instances.

Requirement	
ST 1903.1-17	Metadata producers shall set the MIMD class's version attribute in every MIMD instance to the version specified in MISB ST 1903.

### 7.1.2 Attribute 34 – compositeProductSecurity

*This attribute is a directed association (or reference) to Class Security. Section 7.3 defines the Security class.*

<sup>1</sup> For consistency across the MIMD suite of documents, MISB ST 1901 defines the label designations for the columns in the attribute tables.

The Composite Product Security (compositeProductSecurity) attribute is a reference to the highest-level security class instance for the entire Motion Imagery product (i.e., Motion Imagery, audio, and metadata). Although the security list may include several security levels, the composite security is a reference to the highest-level security class instance that applies to the whole Motion Imagery Product. Section 6.1.5 provides background and details for this attribute. Producers need to set the compositeProductSecurity value in every MIMD packet.

Requirement(s)	
ST 1903-02	Every MIMD instance shall include the compositeProductSecurity attribute.
ST 1903-03	The compositeProductSecurity shall be the reference to the highest-level classification in the security list for all essences within the Motion Imagery product.

### 7.1.3 Attribute 35 – compositeMotionImagerySecurity

*This attribute is a directed association (or reference) to Class Security. Section 7.3 defines the Security class.*

The Composite Motion Imagery Security (compositeMotionImagerySecurity) attribute is the reference to the security class instance for all imagery essences. Producers optionally set this value; if the value is not set or becomes ROC\_Unknown, the compositeProductSecurity is the default. Section 6.1.5 provides background and details for this attribute.

Requirement	
ST 1903-04	Where a producer sets the compositeMotionImagerySecurity, the compositeMotionImagerySecurity shall be the reference to the highest-level classification in the security list for all Motion Imagery essences.

### 7.1.4 Attribute 36 – compositeAudioSecurity

*This attribute is a directed association (or reference) to Class Security. Section 7.3 defines the Security class.*

The Composite Audio Security (compositeAudioSecurity) attribute is the reference to the security class instance for all audio essences. Producers optionally set this value; if the value is not set or becomes ROC\_Unknown, the compositeProductSecurity is the default. Section 6.1.5 provides background and details for this attribute.

Requirement	
ST 1903-05	Where a producer sets the compositeAudioSecurity, the compositeAudioSecurity shall be the reference to the highest-level classification in the security list for all audio essences.

### 7.1.5 Attribute 37 – compositeMetadataSecurity

*This attribute is a directed association (or reference) to Class Security. Section 7.3 defines the Security class.*

The Composite Metadata Security (compositeMetadataSecurity) attribute is the reference to the security class instance for security marking the whole metadata essence. Producers optionally set

this value; if the value is not set or becomes ROC\_Unknown, the compositeProductSecurity is the default. Section 6.1.5 provides background and details for this attribute.

Requirement	
ST 1903-06	Where a producer sets the compositeMetadataSecurity, the compositeMetadataSecurity shall be the reference to the highest-level classification in the security list for all metadata instances.

### 7.1.6 Attribute 38 – timers

*This attribute is a list of Timer classes. Section 7.2 defines the Timer class.*

The MIMD Model requires the list of Timer classes to contain at least one Timer class. Additionally, each Timer needs to meet the requirements for a Level 1 timer, as defined below.

MIMD Timer classes have three levels of timing capability:

- Level 1 is the minimum required where the Timer instance includes the nanoPrecisionTimestamp in every MIMD Packet and the utcLeapSeconds cannot be ROC\_Unknown.
- Level 2 adds uncertainty information by including a non-zero standard deviation value for the nanoPrecisionTimestamp to the Level 1 required attributes.
- Level 3 adds Time Transfer attributes from MISB ST 1603 to the Level 2 attributes. MISB ST 1603 defines these attributes, provides background on Time Transfer metadata, and specifies when they should appear.
- The Timer class attributes include all MISB ST 1603 values; however, the Receptor Clock Uncertainty is the standard deviation of the nanoPrecisionTimeStamp value. The Base Attributes provide the method for specifying the standard deviation for these values.

Requirement	
ST 1903-07	All MIMD Packets shall include, at a minimum, a Level 1 Timer Instance in the Timer list.

### 7.1.7 Attribute 39 – platforms

*This attribute is a list of Platform classes. MISB ST 1905 defines the Platform class.*

### 7.1.8 Attribute 40 – securityOptions

*This attribute is a list of Security classes. Section 7.3 defines the Security class.*

The MIMD Model requires the list of Security classes to contain at least one Security class instance which has its classifyingMethod and classification attribute values in the Known-Changing or Known-Static state (see Report-on-Change in the Motion Imagery Handbook). Therefore, there is always a Security class instance available (with Known-Changing or Known-Static attribute values) for the MIMD class's compositeProductSecurity attribute to reference.

Requirement	
ST 1903-10	The Security class list shall have at least one Security class instance in the Known-Static or Known-Changing state.

### 7.1.9 Attribute 41 – groupIdRange

The groupIdRange attribute specifies the minimum and maximum group identifier in use throughout the MIMD Model. The group identifier is part of the mimdId unique identifier. See MISB ST 1904, Base class attribute mimdId, for description of mimdId, the group identifier and the requirements for the groupIdRange attribute.

The groupIdRange attribute is a two value Tuple with the first value equal to the minimum group identifier and the second value equal to the maximum group identifier.

If the MIMD model instances mimdIds do not use a group value greater than zero (the default group identifier), the groupIdRange attribute does not need to be defined.

### 7.1.10 Attribute 42 – algorithms

*This attribute is a list of Algorithm classes. Section 7.4 defines the Algorithm class.*

The list of algorithms may contain classes which use the same algorithm, but different versions and/or parameters. For example, the algorithms list may include a five copies of the same generalized segmentation algorithm but each algorithm uses a different set of parameters (e.g., “car”, “truck”, ...) or different versions (e.g., for testing and comparison).

Classes within the MIMD model may include an “algorithm” attribute which references an element of the Algorithm list.

### 7.1.11 Attribute 43 – ontologies

*This attribute is a list of Ontology classes. Section 7.5 defines the Ontology class.*

### 7.1.12 Attribute 44 – tracks

*This attribute is a list of Track classes. MISB ST 1907 defines the Track class.*

## 7.2 Timer Class

The Timer class includes attributes for reporting time and providing time transfer synchronization data. Refer to the Motion Imagery Handbook and MISB ST 1603 for background and descriptions of Time Transfer attributes.

The MIMD Timer class contains the attributes listed in Table 2.

**Table 2: MIMD Timer Class Attributes<sup>2</sup>**

<b>Id</b>	<b>Name</b>	<b>Type</b>	<b>Min</b>	<b>Max</b>	<b>Res</b>	<b>MLen</b>	<b>Units</b>	<b>Ref</b>
33	nanoPrecisionTimestamp	UInt	--	--	N/A	N/A	ns	7.2.1
34	utcLeapSeconds	Integer	--	--	N/A	N/A	s	7.2.2
35	timeTransferMethod	TimeTransferMethod	N/A	N/A	N/A	N/A	None	7.2.3
36	correctionMethod	CorrectionMethod	N/A	N/A	N/A	N/A	None	7.2.4
37	referenceSource	ReferenceSource	N/A	N/A	N/A	N/A	None	7.2.5
38	syncPulseFreq	Real	0.0	--	--	N/A	Hz	7.2.6
39	unlockTime	UInt	--	--	N/A	N/A	ns	7.2.7
40	lastSyncDiff	UInt	--	--	N/A	N/A	ns	7.2.8
41	driftRate	Real	--	--	--	N/A	ns/s	7.2.9
42	signalSourceDelay	UInt	--	--	N/A	N/A	ns	7.2.10
43	source	REF<Stage>	N/A	N/A	N/A	N/A	None	7.2.11
44	correctionOffset	Integer	--	--	N/A	N/A	ns	7.2.12

### 7.2.1 Attribute 33 – nanoPrecisionTimestamp

MISB ST 0603 [9] defines the Nano Precision Time Stamp (nanoPrecisionTimestamp), which is an unsigned integer representing time measured from the MISP Epoch (see Motion Imagery Handbook) in nanoseconds. The nanoPrecisionTimestamp is a required element in every Timer instance.

Requirement	
ST 1903-08	Every Timer instance shall include a nanoPrecisionTimestamp attribute value.

### 7.2.2 Attribute 34 – utcLeapSeconds

The Time Transfer LS Item 2 in MISB ST 1603 defines the UTC Leap Second Offset (utcLeapSeconds) and usage rules. The utcLeapSeconds cannot be ROC\_Unknown.

Requirement	
ST 1903-09	The utcLeapSeconds attribute value shall never be ROC_Unknown.

### 7.2.3 Attribute 35 – timeTransferMethod

*This attribute is an enumeration. Section 8.1 defines the TimeTransferMethod enumeration.*

Note: The timeTransferMethod does not include a “Free-Wheeling” enumeration value when the reference clock sync is lost. When sync is lost the unlockTime attribute provides the time since the last lock – refer to MISB ST 1603 for further details.

<sup>2</sup> For consistency across the MIMD suite of documents, MISB ST 1901 defines the label designations for the columns in the attribute tables.

#### **7.2.4 Attribute 36 – correctionMethod**

*This attribute is an enumeration. Section 8.2 defines the CorrectionMethod enumeration.*

#### **7.2.5 Attribute 37 – referenceSource**

*This attribute is an enumeration. Section 8.3 defines the ReferenceSource enumeration.*

#### **7.2.6 Attribute 38 – syncPulseFreq**

The Time Transfer LS Item 4 in MISB ST 1603 defines the Synchronization Pulse Frequency (syncPulseFreq).

#### **7.2.7 Attribute 39 – unlockTime**

The Time Transfer LS Item 5 in MISB ST 1603 defines the Unlock Time (unlockTime).

#### **7.2.8 Attribute 40 – lastSyncDiff**

The Time Transfer LS Item 6 in MISB ST 1603 defines the Last Synchronization Difference (lastSyncDiff).

#### **7.2.9 Attribute 41 – driftRate**

The Drift Rate (driftRate) attribute is defined the same as in the Time Transfer LS Item 7, Drift Rate, in MISB ST 1603; however, in the Timer class it is measured in nanoseconds per second (ns/s).

#### **7.2.10 Attribute 42 – signalSourceDelay**

The Time Transfer LS Item 8 in MISB ST 1603 defines the Signal Source Delay (signalSourceDelay).

#### **7.2.11 Attribute 43 – source**

*This attribute is a directed association (or reference) to Class Stage. MISB ST 1906 defines the Stage class.*

The source value is a reference to a navigation Source class in MISB ST 1906's Staging system. This allows the timer to be associated with a physical device generating the timing information, e.g., a NAVSAT system such as GPS.

#### **7.2.12 Attribute 44 – correctionOffset**

The UAS Datalink LS Item 137 in MISB ST 0601 [10] defines the Correction Offset (correctionOffset).

## 7.3 Security Class

Table 3 lists the attributes for the Security Class.

**Table 3: MIMD Security Class Attributes<sup>3</sup>**

Id	Name	Type	Min	Max	Res	MLen	Units	Ref
33	classifyingMethod	String	N/A	N/A	N/A	100	None	7.3.1
34	classification	String	N/A	N/A	N/A	500	None	7.3.2

All instances of the Security class must define a classification.

Requirement(s)	
ST 1903-11	Every Security class instance's classification attribute shall never be in the ROC_Unknown state.
ST 1903-12	Every Security class instance's classifyingMethod attribute shall never be in the ROC_Unknown state.

### 7.3.1 Attribute 33 – classifyingMethod

The classifying method (classifyingMethod) attribute (see Table 4) defines the method for parsing the classification attribute.

**Table 4: MIMD Classifying Method**

Method Name	Meaning	Reference
US-1	United States Classification (DoDM 5200.01 Volume 2 [11])	Section 7.3.1.1

Requirement	
ST 1903-13	The classifyingMethod attribute shall only contain values from the Method Name column of MISB ST 1903 Table 4.

#### 7.3.1.1 United States Classification Method

The United States Classification method uses the rules defined in DoDM 5200 Volume 2. DoDM 5200 Volume 2, Enclosure 4 (Marking Standard) defines the string syntax for security marking the classification attribute uses. For convenience, the string syntax below duplicates the syntax from DoDM 5200 Volume 2. If any discrepancies exist between DoDM 5200 Volume 2 and the information below, refer to DoDM 5200 Volume 2 as the definitive source.

<sup>3</sup> For consistency across the MIMD suite of documents, MISB ST 1901 defines the label designations for the columns in the attribute tables.

**CLASSIFICATION//SCI/SCI-SUBCONTROL//SAP//AEA//FGI//DISSEM/DISSEM//OTHER DISSEM**

Where:

Keyword	Meaning or valid values
Classification	U.S. Classification, Foreign Government Classification, or Joint Classification
SCI	SCI Control System
SCI-SUBCONTROL	SCI Sub-Control System
SAP	Special Access Program
AEA	Atomic Energy Act Info
FGI	Foreign Government Information
DISSEM	Dissemination Controls
OTHER DISSEM	Other Dissemination Control Markings

Separators:

Separator	Meaning
//	Use a double forward slash to separate marking categories
/	Use a single forward slash to separate multiple values within a marking category
-	Use a hyphen to link a marking to a sub-marking (e.g., SI-G or RD-SIGMA)
“ ”	Use a space to separate multiple sub-markings and multiple trigraph or tetragraph codes in the FGI Marking (e.g., //SI-ABC-G XYZW YYYY// or //FGI GBR JPN//
,	Use a comma to separate multiple trigraph or tetragraph codes in the REL TO marking

### 7.3.2 Attribute 34 – classification

The classification string is a classification definition following the classifyingMethod rules.

## 7.4 Algorithm Class

The Algorithm class identifies and describes a process used for computing values within the MIMD Model. MIMD classes which reference this class provide additional details about the context of the algorithm.

**Table 5: Algorithm Class Attributes**

Id	Name	Type	Min	Max	Res	MLen	Units	Ref
33	name	String	N/A	N/A	N/A	100	None	7.4.1
34	version	String	N/A	N/A	N/A	50	None	7.4.2
35	type	String	N/A	N/A	N/A	100	None	7.4.3
36	parameters	String	N/A	N/A	N/A	100	None	7.4.4

### 7.4.1 Attribute 33 – name

The name attribute contains the name of the algorithm, independent of the version number. For example, if an algorithm for processing data is the “ACME Quick Sort v2.7.1”, the resulting value in the name attribute is “ACME Quick Sort”.

Requirement	
ST 1903.1-14	Every Algorithm instance shall define the name attribute.

### 7.4.2 Attribute 34 – version

The version attribute contains the version for the algorithm, independent of the algorithm name. For example, if an algorithm for processing data is the “ACME Quick Sort v2.7.1”, the resulting value in the version attribute is “2.7.1”.

### 7.4.3 Attribute 35 – type

The type attribute is the category of the algorithm, e.g., Sorting, Filter, Segmentation. For example, if an algorithm for processing data is the “ACME Quick Sort v2.7.1”, the resulting value in the type attribute could be “Sorting”.

### 7.4.4 Attribute 36 – parameters

The parameters attribute contains the algorithm parameters in use for this algorithm. For example, if a generalized segmentation algorithm is the “ACME Segmentation v3.1.4,” and when it is run (for this Algorithm instance) the parameters are to segment “cars” from imagery, the parameters attribute is set to “car”, or whatever the application needs to specify “cars”.

## 7.5 Ontology Class

The Ontology class describes the [grouping] class or type of a physical object (aircraft, watercraft, car, truck, train, dismount, etc.) to an arbitrary level of detail.

**Table 6: Ontology Class Attributes**

Id	Name	Type	Min	Max	Res	MLen	Units	Ref
33	ontology	String	N/A	N/A	N/A	100	None	7.5.1

### 7.5.1 Attribute 33 – ontology

The Ontology attribute is a Uniform Resource Identifier (URI). Refer to the Web Ontology Language (OWL) [12] for the ontology specification.

The Jet Propulsion Laboratory Semantic Web for Earth and Environmental Terminology (SWEET) (<https://bioportal.bioontology.org/ontologies/SWEET>) provides a collection of ontologies, written in the OWL ontology language which serves as examples and starting points for the development of additional domain-specific extended ontologies.

Requirement(s)
----------------

ST 1903.1-15	The ontology shall be expressed using the OWL Web Ontology Language.
ST 1903.1-16	All instances of the Ontology Class shall contain the ontology attribute.

## 8 Model Enumerations

### 8.1 TimeTransferMethod Enumeration

The Time Transfer LS Item 3 in MISB ST 1603 defines all the Time Transfer Parameters, which includes the Time Transfer Method Enumeration.

**Table 7: Enumeration Values for TimeTransferMethod**

Id	Name	Description
0	Unknown	Unknown Time Transfer Method
1	GPS	Global Positioning System (GPS) PPS
2	PTP_V1	Precision Time Protocol (PTP) - Version 1
3	PTP_V2	Precision Time Protocol (PTP) - Version 2
4	NTP_V3_3	Network Time Protocol (NTP) - Version 3.3
5	NTP_V4	Network Time Protocol (NTP) - Version 4
6	IRIG_A	Inter-range Instrumentation Group (IRIG-A)
7	IRIG_B	Inter-range Instrumentation Group (IRIG-B)

For more information on various Time Transfer Methods, refer to:

- Global Positioning System (GPS) PPS [13]
- Precision Time Protocol (PTP) - Version 1 [14]
- Precision Time Protocol (PTP) – Version 2 [15]
- Network Time Protocol (NTP V3.3) [16]
- Network Time Protocol (NTP V4) [17]
- Inter-range Instrumentation Group (IRIG-A or IRIG-B) [18]

### 8.2 CorrectionMethod Enumeration

The Time Transfer LS Item 3 in MISB ST 1603 defines the Time Transfer Parameters, which includes the Correction Method Enumeration.

**Table 8: Enumeration Values for CorrectionMethod**

Id	Name	Description
0	Unknown	Unknown Correction Method
1	Jam	Jam Correction Method
2	Slew	Slew Correction Method

### 8.3 ReferenceSource Enumeration

The Time Transfer LS Item 3 in MISB ST 1603 defines the Time Transfer Parameters, which includes the Reference Source Enumeration.

**Table 9: Enumeration Values for ReferenceSource**

Id	Name	Description
0	Unknown	Reference Source status is unknown
1	Not_Synchronized	Reference Source is not synchronized to an atomic source
2	Synchronized	Reference Source is synchronized to an atomic source

## Appendix A      Deprecated Requirements

Requirement	
ST 1903-01 (Deprecated)	Metadata producers shall set the MIMD class's versionNumber attribute in every MIMD instance to the minor version of the MISB ST 1903 document the producer has implemented.